

# 이중분기 확장을 통한 등치선 삼각화의 다중 분기 알고리즘

(A Multiple Branching Algorithm of Contour Triangulation by Cascading Double Branching Method)

최 영 규 <sup>†</sup>

(Young-Kyu Choi)

**요 약** 등치선(wire-frame contour)으로 표현된 물체의 볼륨정보에서부터 3차원 표면을 재구성하는 방법을 제안한다. 등치선 삼각화법(contour triangulation)이라고도 하는 이 방법에서 가장 문제가 되는 것이 인접 단층사이에서 표면이 분기하는 경우에 발생하는데, 이것은 하나의 등치선이 인접한 층의 두 개의 상의 등치선과 연결되는 형태로 나타나며, 표면 생성시의 많은 모호성을 발생시킨다. 본 논문에서는 이러한 분기문제를 가장 일반적으로 발생하는 이중분기문제와 그 이상의 다중분기문제로 구분하고, 먼저 이중분기 알고리즘을 제안하였으며, 다중분기문제를 다수의 이중분기문제로 단순화하는 다중분기 알고리즘을 제안하였다. 제안된 이중분기 알고리즘은 모 등치선을 분할하는 방법을 이용하였는데, 먼저 해협다각형을 정의하고 이를 삼각분할하여 분할선을 구하는 것에 바탕을 두고 있다. 이 방법은 이중분기가 매우 복잡하게 나타나는 경우에도 잘 적용이 되며, 분할선의 레벨을 조절함으로써 매우 사실적인 표면을 만들어 낼 수 있다는 장점이 있다. 또한 다중분기문제를 단층 간격의 문제로 규정하고, 인접한 두 층 사이에 가상의 등치선을 추가하여 가지 등치선을 연속적으로 병합하는 방법으로 해결하였다. 제안된 방법은 등치선 삼각화의 가장 큰 문제인 분기문제를 해결하기 위한 매우 구조적인 접근방법으로, 다양한 실제 등치선 데이터에 적용한 결과 좋은 성능을 나타냈다.

**Abstract** This paper addresses a new triangulation method for constructing surface model from a set of wire-frame contours. The most important problem of contour triangulation is the branching problem, and we provide a new solution for the double branching problem, which occurs frequently in real data. The multiple branching problem is treated as a set of double branchings and an algorithm based on contour merging is developed. Our double branching algorithm is based on partitioning of root contour by Toussiant's polygon triangulation algorithm[14]. Our double branching algorithm produces quite natural surface model even if the branch contours are very complicate in shape. We treat the multiple branching problem as a problem of coarse section sampling in z-direction, and provide a new multiple branching algorithm which iteratively merge a pair of branch contours using imaginary interpolating contours. Our method is a natural and systematic solution for the general branching problem of contour triangulation. The result shows that our method works well even though there are many complicated branches in the object.

## 1. 서 론

3차원 물체의 표면정보는 의료영상이나 컴퓨터그래픽

스 등 다양한 분야에서 매우 중요하게 사용되고 있다. 이러한 정보들은 흔히 단층촬영 영상으로 주어지게 되는데, 단층 영상에서부터 그 안에 포함된 볼륨 정보를 가시화하기 위하여 다양한 방법들이 제안되고 있다. 이들을 3차원 영상을 생성하는 렌더링의 관점에서 보면, 영상정보에서 표면모델을 먼저 추출한 후 이를 렌더링하여 3차원 영상을 만드는 *표면기반(surface-based)* 렌

<sup>†</sup> 정 회 원 : 한국기술교육대학교 정보기술공학부 교수  
ykchoi@kut.ac.kr

논문접수 : 1999년 7월 19일

심사완료 : 1999년 11월 30일

더링방법[3-13]과, 표면모델을 생성하지 않고 볼륨 데이터에서 직접 영상을 얻어내는 볼륨기반(volume-based) 렌더링방법[1,2]으로 크게 나눌 수 있는데, 본 논문에서는 가시화뿐 아니라 체적의 계산이나 모의 수술 등의 목적으로도 사용할 수 있는 표면기반 방법에 초점을 맞춘다.

표면기반 렌더링방법은 모델링시 입력의 종류에 따라 다시 두 가지로 나눌 수 있는데, 공간이 복셀이라고 하는 정육면체 체적소로 균일하게 나뉘어져 있다고 가정하고 이들 사이에서 표면을 만들고자 하는 복셀(voxel) 기법이라고도 하는 체적 재구성법(volume reconstruction)[3-6]과, 각 단층면과 물체의 표면이 만나는 등치선(等値線, contour, 같은 값의 밀도면(isodensity surface)과 단층면이 만나는 선이라는 의미)으로 부터 이들 사이의 영역을 삼각형 표면조각으로 채우는 표면 재구성(surface reconstruction)법[7-13]으로 나눌 수 있다. 체적 재구성은 일반적으로 알고리즘이 간단하며, 다양한 예외사항에도 비교적 강건하게 수행된다는 장점이 있다. 이에 비하여, 표면 재구성은 등치선 삼각분할법(contour triangulation)이라고도 하는데, 등치선이 선형으로 보간되어 있으므로 추가적인 anti-aliasing 과정이 필요 없고, 체적 재구성법에서 발생하는 거친 표면(jagged surface)처리가 필요 없으며, 각 단층에 대한 영상을 저장할 필요가 없이 추출된 경계선만을 이용하여 모델링 할 수 있으므로 데이터양이 적다는 장점을 가지고 있다. 또한 등치선에 직선 근사화법 등을 적용하여 근사화 할 수 있어 체적에 근거한 방법에 비하여 생성되는 표면조각의 수가 훨씬 적으며, 평활 명암 법이나 Phong 명암법 등을 적용하여 가시화 하거나, Ray-casting법 등을 적용하여 고화질의 3차원 영상을 만들어 낼 수 있다. 이러한 장점에도 불구하고 표면 재구성은 입력 등치선의 형태에 따라 경우에 따라서는 매우 부자연스러운 표면을 만들 수 있다는 단점이 있다. 이것은 특히 표면이 여러 가닥으로 복잡하게 분기되는 경우에 매우 심각하게 발생하게 된다. 본 논문에서는 이러한 등치선 삼각 분할법에 의한 표면 재구성 방법을 제안한다. 특히 이 방법의 가장 중요한 문제인 표면 분기를 체계적으로 해결할 수 있는 방법을 제안한다.

## 2. 등치선 삼각분할의 개요

그림 1은 단층 영상으로부터 등치선 삼각분할에 의해 물체의 표면모델을 재구성하는 전체과정을 보여주었다. (a)에서와 같이 각 단층에서의 등치선이 주어지면, 등치선간의 연결성 결정해야 하며(b), 인접 등치선의 연

결상태에서 분기가 발생하면 이를 처리해야 하고(c), 마지막으로 인접한 층의 한 쌍의 등치선사이의 표면을 근사(approximation)하는 삼각형 표면조각을 생성함으로써(d) 표면 모델이 구해지게 된다.

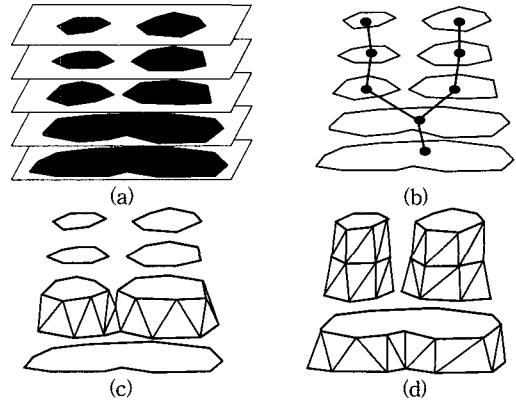


그림 1 등치선 삼각분할에 의한 표면 재구성 단계

### 2.1 등치선의 연결상태 결정(Correspondence Problem)

각 단층에서 등치선들이 구해지게 되면 이들 사이를 표면조각들로 채우기 전에 먼저 어느 등치선이 인접한 층에 있는 어느 등치선과 표면으로 연결되어야 하는가를 결정해야 하는데, 이것을 등치선 대응문제라고 한다. NMR 등에 의해 얻어지는 영상들은 일반적으로 단층촬영의 층과 층 사이가 좁은데 이러한 경우에는 보통 중첩성을 이용하여 인접한 층에 있는 두 등치선들이 서로 연결되어야 하는지 아닌지를 결정하게 된다. 즉, 두 등치선을 단층면과 수직방향으로 투사(projection)하였을 때, 두 등치선의 내부 영역들이 중첩되는 정도가 전체 면적에 대하여 임계치 이상으로 큰 경우, 이들은 연결되어야 한다고 판단하게 된다. 응용분야에 따라서는 이러한 중첩성만으로는 등치선간의 연결성 문제가 해결되지 않는 경우가 발생한다. 이것은 인체내의 동맥이나 정맥등과 같은 혈관을 모델링 할 때와 같이, 단층사이의 거리가 비교적 멀고 모델링 하고자 하는 대상체의 크기가 비교적 작은 경우에 흔히 발생하게 된다. 이러한 경우에는, 연결이 먼저 결정된 등치선들을 실린더로 모델링 하고 이들을 성장시키는 방법(cylinder growing)을 사용하는 등의 여러 가지 휴리스틱을 적용하여야 하는 경우도 있다[12]. 본 논문에서는 중첩성을 이용하여 등치선간의 연결상태를 결정하였다.

### 2.2 등치선간의 표면 생성(Tiling Problem)

등치선 사이의 연결성이 결정되면 인접한 층에 있는 두 등치선 사이의 물체 표면을 삼각형 표면조각들로 근사하는 과정이 이어지는데 이것을 표면조각 생성 문제(tiling problem)이라 한다. 이러한 등치선사이의 표면조각 생성문제는 토로이달 그래프(Toroidal graph)[7]에 의해 잘 표현되는데, 인접한 등치선 사이의 하나의 삼각형 표면조각은 그래프의 하나의 간선(edge)으로 대응되며 결과적으로 이 그래프의 하나의 경로(path)는 두 등치선 사이의 하나의 가능한 표면 구성에 해당하게 된다. 이러한 표면 생성문제를 해결하기 위한 방법들은 크게 최적해(optimal) 방법과 경험적(heuristic) 방법으로 크게 나눌 수 있다. 최적해 방법[7]은 주어진 문제를 Toroidal 그래프의 탐색 문제로 변환하고 주어진 비용함수에 대해 최적의 경로를 찾아내는 방법이다. 이에 비하여 경험적 방법[8-11]은 등치선의 국부적인 탐색으로 삼각형 조각을 생성한다. 최적해 방법이 그래프의 전역 탐색에 의해 이루어지므로 계산량이 많아지는데 비하여 이 방법은 국부적인 정보들만을 이용하여 삼각형 조각을 생성해 내므로 계산량이 훨씬 적다. 또 결과적으로 얻어지는 삼각형 조각들이 "최적"의 의미는 없지만 시각적으로는 타당하게 보이는 경우가 많으므로 많은 사람들에게 의해 연구되어 왔다. 예를 들어 Christiansen[8]은 두 개의 가능한 삼각형 조각 중 변의 길이가 작은 것을 연속적으로 선택하는 방법을 제안하였고, Gapanathy[9]는 두 등치선의 길이를 먼저 정규화시키고 시작점에서부터의 거리가 가능한 한 비슷하게 유지하면서 표면을 얻는 방법을 제시하였다. Ekoule[10]은 등고선을 블록 다각형으로 계속 분할하는 방법으로 표면조각들을 구하였으며, 본 논문에서는 등치선 표면생성문제를 연속적인 체인 분할문제로 단순화하는 것에 기반으로 하는 알고리즘[11]을 이용하여 표면모델을 구하였다.

### 2.3 표면 분기(Branching Problem)

그림 1의 (c)의 둘째 층과 셋째 층 사이와 같이 아래 층에 있는 하나의 등치선이 위층의 두 개 이상의 등치선으로 분기되는 경우 표면을 생성하는데 모호성이 발생하게 된다. 일반적인 접근방식은 분기가 발생하는 가지 등치선들의 정보를 이용하여 하나의 가상의 등치선을 두 층 사이의 가상의 층에 만들고 이를 이용하여 아래층과 위층의 등치선을 연결하는 것이다[8,11]. 이 가상의 등치선으로는 가지 등치선의 무게중심들의 볼록 껍질(convex hull)[11]이 많이 사용된다. 또 Meyers[13]은 이중분기에 대한 해결책을 제안했는데, 가지 등고선 사이를 계곡(canyon)이라 정의하고 그 계곡의 바닥을

삼각형 표면으로 채워서 먼저 표면 모델을 얻고 후처리 과정에서 계곡의 골을 만드는 방법을 제안하였는데, 이것을 다중분기로 확장할 수 있는 방법을 제시하지 못하였다. 또 Boissonnat[13]은 Voronoi Diagram을 이용하여 이 문제를 해결하고자 하였다. 이들 알고리즘은 이중분기에만 적용되거나 모든 분기문제를 동일하게 보고 처리하는 방법을 사용하였다. 일반적으로 단층간의 거리가 물체의 크기에 비하여 충분히 짧을 경우에, 대부분의 분기문제는 이중분기로 나타나게 되는데, 이러한 개념으로 다중분기를 보면, 다중분기는 인접한 단층사이의 거리가 멀어서 생긴 문제라고 생각할 수 있다. 따라서 본 논문에서는 분기 문제를 가장 흔히 발생하는 이중분기 문제와 3중 이상의 다중분기문제로 나누고, 이중분기를 해결하기 위한 알고리즘을 제안하며, 다중분기문제를 연속적인 이중분기문제로 단순화하여 해결하는 알고리즘을 제안한다. 이런 접근방법은 다양한 분기에 대해 기존의 획일화된 알고리즘들에 비해 훨씬 자연스러운 표면 모델을 만들 수 있다.

### 3. 용어 정리

이 장에서는 본 논문에서 사용되는 용어 및 표기를 정의한다.

**정의 1:** 단층의 레벨(level)은 그 단층의 높이를 나타내고  $z_k$  로 표기한다.  $z_k$  와  $z_{k+1}$ 은 인접한 단층의 레벨이며,  $z_k$ 가 하위레벨이다.

**정의 2:** 등치선(contour)은 평면상의 임의의 단순 다각형(simple polygon)으로 정의되며  $C_i^*$  로 표기된다.  $C_i^*$  는 레벨이  $z_k$  인 단층의  $i$  번째 등치선이며, 정점  $(v_0, v_1, v_2, \dots, v_{n-1})$ 으로 구성된다. 이때  $n$ 은 등치선의 총 정점의 수이며, 정점들은 반 시계 방향으로 정렬되어 있다.

**정의 3:** 단일 분기(single branching)는 레벨  $z_k$  인 단층의 등치선  $C_i^*$  가, 레벨  $z_{k+1}$  인 단층의 오직 하나의 등치선  $C_j^{*+1}$  연결되어 있는 경우이며,  $\langle (C_i^*), (C_j^{*+1}) \rangle$  로 표기한다.

**정의 4:** 이중 분기(double branching)는 레벨  $z_k$  인 단층의 등치선  $C_i^*$  가, 레벨  $z_{k+1}$  인 단층의 두개의 등치선  $C_j^{*+1}$  및  $C_l^{*+1}$ 와 함께 연결되어 있는 경우이며,  $\langle (C_i^*), (C_j^{*+1}), (C_l^{*+1}) \rangle$  로 표기한다. 이때  $C_i^*$  를 모 등치선(root contour)이라 하고  $C_j^{*+1}, C_l^{*+1}$ 를 가지 등치선(branch contour)라 한다.

**정의 5:** 다중 분기(multiple branching)는 레벨  $z_k$

인 단층의 등치선  $C_i^k$  가, 레벨  $z_{k+1}$  인 단층의  $N$  개의 등치선  $C_1^{k+1}, C_2^{k+1}, \dots, C_N^{k+1}$  과 함께 연결되어 있는 경우이며,  $\langle (C_i^k), (C_1^{k+1}, \dots, C_N^{k+1}) \rangle$  로 표기한다. 이때  $N \geq 2$  이며,  $C_i^k$  를 모 등치선,  $C_1^{k+1}, \dots, C_N^{k+1}$  를 가지 등치선이라 한다.

그림 2는 다양한 인접한 레벨의 다양한 등치선 분기 예를 나타내고 있는데, (a)는 단일분기이며, (b)는 이중분기, (d)는 다중분기를 나타내고 있다. 또한 (c)는 이중분기가 인접한 단층에서 여러번 나타나 있는 복잡한 분기문제를 나타내고 있다.

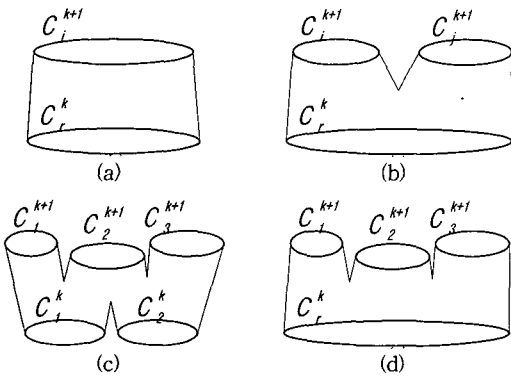


그림 2 다양한 표면 분기 예

4. 이중 분기

등치선으로 표현된 물체가 복잡한 형태이면 다양한 표면 분기를 수반하게 된다. 그 중에서 가장 흔히 일어나는 형태가 이중분기이다. 분기문제는 각 인접한 단층 사이의 거리에 영향을 받게 된다. 즉 물체의 크기에 비하여 단층사이의 거리가 이 큰 경우에는 인접한 단층간에 동질성이 상대적으로 떨어지고, 따라서 다중 분기가 생길 가능성이 많아진다. 하지만 단층 간격이 상대적으로 작은 경우에는 등고선 사이의 동질성이 많아지게 되고, 인접한 레벨의 등치선들 사이의 표면은 다중분기로 이어지기보다는 이중분기의 형태로 표면이 연결될 가능성이 많다. 따라서 본 논문에서는 다중분기와 이중분기를 구분하였고, 가장 흔히 발생하는 이중분기에 대해 먼저 해결책을 제시한다.

이중 분기는 한 레벨의 임의의 모 등치선이 인접 레벨의 두 개의 가지 등치선과 표면으로 연결되는 경우이다. 본 논문에서 제안하는 이중분기 해결방법은 가지 등치선 정보를 이용하여 모 등치선을 두 개로 분할하는

것에 기초하고 있다. 만약 모 등치선이 적절히 분할되면, 원래의 이중분기 문제가 두 개의 단일분기 문제로 간략화 된다. 즉 모 등치선의 분할 등치선을 각각 하나의 가지 등치선에 연결하는 방법으로 이중 분기시 닫힌 표면을 만들게 된다. 예를 들면, 그림 3에서 레벨  $z_k$  에 있는 모 등치선  $C_i^k$  가 레벨  $z_{k+1}$  에 있는 가지 등치선  $C_i^{k+1}$  와  $C_j^{k+1}$  로 분기되는 이중분기  $\langle (C_i^k), (C_i^{k+1}, C_j^{k+1}) \rangle$  가 두 개의 단일분기  $\langle (C_{rs1}^k), (C_i^{k+1}) \rangle$ ,  $\langle (C_{rs2}^k), (C_j^{k+1}) \rangle$  문제로 간략화 된다.

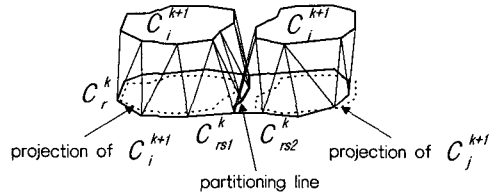


그림 3 이중분기의 간략화

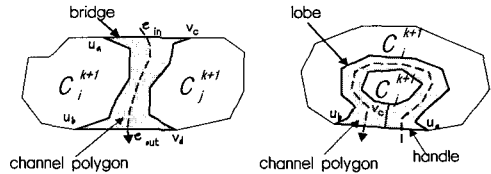


그림 4 해협(channel polygon)의 정의

4.1 해협 다각형(Channel Polygon) 정의

이중분기  $\langle (C_i^k), (C_i^{k+1}, C_j^{k+1}) \rangle$  가 주어졌을 때, 모 등치선의 자연스러운 분할이 가장 중요한 과제가 된다. 분할을 위한 가장 중요한 정보는 그림 3의 점선과 같이 두 가지 등치선을 모 등치선 레벨에 수직 사상(vertical projection)한 등치선이 될 수 있다. 분할 선(partition line)이 이들 사이에 있으면 결과적으로 생기는 표면이 어느 한쪽의 가지 등치선에 편중되지 않고 균일하게 나타나기 때문이다. 이렇게 분할선을 구하기 위해 먼저 해협(channel polygon)을 정의하였는데, 그림 4에서 해협의 예를 보여주고 있다. 해협은 두 가지 등치선의 볼록 꺾질  $CH(C_i^{k+1}, C_j^{k+1})$ 을 먼저 구하고, 이것의 간선들 중에서 다리(bridge)이나 핸들(handle)로부터 정의된다. 즉 그림 4의 왼쪽처럼 볼록 꺾질에 다리가 두 개 나타나는 경우, 이를 포함하고 두 가지 등치선의 가까운 부분의 간선들을 이용하여 그림의 회색부분과 같은 가지 등치

선간의 해협을 정의한다. 그림 4의 오른쪽은 한 등치선이 다른 하나의 볼록 꺾질에 완전히 포함되는 경우로 이때에는 핸들과 가장 가까운 정점( $v_c$ )을 이용하여 그림과 같이 해협을 정의하게 되는데, 이와 같이 정의된 해협은 모 등치선의 분할선이 지나갈 영역이 된다.

**4.2 해협 다각형(Channel Polygon)으로부터 T-axis추출**

해협으로부터 분할선을 구하는 방법은 여러 가지 알고리즘을 적용할 수 있지만, 다각형의 골격(skeleton)이나 중앙축(medial axis)[16]이 가장 적합한 구조가 된다. 본 논문에서는 해협의 골격으로 T-skeleton을 정의하였다. 이를 구하기 위해 먼저 해협을 Toussaint이 제안한 다각형 분할 알고리즘[14]을 이용하여 삼각 분할한다. 이 알고리즘(sleeve-searching algorithm)은 형태가 복잡하지 않은 단순다각형의 삼각분할에 특히 효과적인 알고리즘으로, 다각형의 형태의 복잡도가 삼각분할의 시간복잡도에 영향을 미치는 알고리즘이다. 이 알고리즘의 시간복잡도는  $O(n(1+t_0))$ 인데,  $n$ 은 정점의 수를 나타내고,  $t_0$ 는 분할된 삼각형중에서 어느 한 변도 다각형의 간선이 아닌 삼각형의 수를 의미한다. 즉  $t_0$ 가 크면 그만큼 복잡한 다각형이 되는데, 일반적으로 해협은 그림 4와 같이 비교적 간단한 형태가 되므로 이 알고리즘에 의한 해협 삼각분할은 실제로 거의 선형시간만에 수행된다. 그림 5의 왼쪽은 이러한 방법으로 해협을 삼각분할 한 결과로, 이 경우는  $t_0=0$ 이다. 해협이 분할되면, 간선  $e_{in}$ 의 중심점에서부터  $e_{out}$ 까지 연속적으로 인접한 대각선(두 정점이 각각 다른 등치선에 속한

간선)의 중심을 연결하면 해협의 골격이 나타나게 되고 이를 T-skeleton이라 정의한다. 그림 5는 해협을 삼각 분할 하고 이를 이용하여 T-skeleton을 구하는 예를 나타내고 있는데, 이를 이용하여 분할선(partitioning line)을 구해 모 등치선을 분할하게 된다.

**4.3 모 등치선의 분할**

T-skeleton이 구해지면 이를 이용하여 그림 6와 같이 모 등치선  $C_i^k$ 의 분할선을 구한다. T-skeleton의 정점들 중에서 모 등치선 외부에 있는 정점들은 먼저 없애고, 나머지 정점 체인의 양끝에 있는 정점과 가장 인접한 정점을 모 등치선에서 찾음으로써 분할선이 만들어진다. 이 분할선은 모 등치선  $C_i^k$ 를 두 개의 부분 등치선  $C_{in}^k$ 과  $C_{out}^k$ 로 분할하게 되며, 결국 원래의 이중분기 문제  $\langle\langle C_i^k, C_j^{k+1} \rangle\rangle$ 가 두 개의 단일분기  $\langle\langle C_{in}^k, C_i^{k+1} \rangle\rangle$  및  $\langle\langle C_{out}^k, C_j^{k+1} \rangle\rangle$ 문제로 단순화되었다.

**4.4 이중분기 알고리즘**

그림 7은 이중분기 알고리즘을 보여주고 있다.

**Theorem 1:** 알고리즘 Double\_Branching의 시간복잡도는  $O((n_i+n_j)(1+t_0+n_r))$ 이다. 이때  $n_i, n_j$  및  $n_r$ 은 각각 두 가지등치선 및 모 등치선의 정점의 수이며,  $t_0$ 는 Toussiant의 알고리즘에서 어느 한 변도 다각형의 간선이 아닌 삼각형의 수를 의미한다.

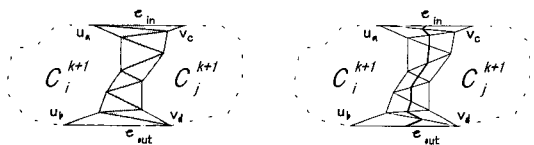


그림 5 T-skeleton의 추출

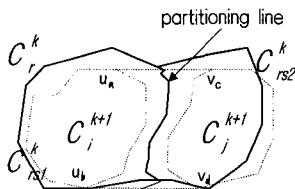


그림 6 모 등치선의 분할

**Procedure Double\_Branching**  
 Input: A root contour  $C_i^k$  at level  $z_k$  and its two branch contour  $C_i^{k+1}$  and  $C_j^{k+1}$  at level  $z_{k+1}$ .  
 Output: Two single branchings:  $\langle\langle C_{in}^k, C_i^{k+1} \rangle\rangle$  and  $\langle\langle C_{out}^k, C_j^{k+1} \rangle\rangle$ .

**Begin**

- [1] Find the channel polygon.
- [2] Find the T-skeleton of the channel polygon.
  - [2.1] Triangulate the channel polygon using the Toussiant's algorithm.
  - [2.2] Extract T-skeleton from the triangulation result.
- [3] Remove exterior vertices in T skeleton.
- [4] Find the closest vertices of the root contour from the two end point of the T-skeleton. The resulting vertices chain is the Partition line.
- [5] Using the partitioning line, divide the root contour  $C_i^k$  into two sub\_contours  $C_{in}^k$  and  $C_{out}^k$ .
- [6] Determine the single branchings:  $\langle\langle C_{in}^k, C_i^{k+1} \rangle\rangle$  and  $\langle\langle C_{out}^k, C_j^{k+1} \rangle\rangle$ .
- [7] Return  $\langle\langle C_{in}^k, C_i^{k+1} \rangle\rangle$  and  $\langle\langle C_{out}^k, C_j^{k+1} \rangle\rangle$ .

**End**

그림 7 이중분기 알고리즘

**Proof:** 단순 다각형의 볼록꺾질의 시간복잡도는  $O(n)$ 이며, 두 단순다각형의 합인 볼록꺾질은 각 단순다

각형의 블록집질의 합은 블록집질과 같다. 또한 두 블록 다각형의 블록집질은 정점수의 합에 비례하는 시간에 구해지므로[17] Step 1의 시간 복잡도는  $\alpha(n_i + n_j)$ 이다. Step 2.1은  $\alpha((n_i + n_j)(1 + t_0))$ 에 수행되고 ([14] 참조) 2.2는  $\alpha(n_i + n_j)$ 에 수행된다. Step 3은 최대  $\alpha(n_i + n_j n_i)$ 만에 수행되며, Step 4와 5는 각각  $\alpha(n_i)$ 시간안에 처리되고 Step 6은  $\alpha(1)$ 시간에 처리된다. 따라서 이 알고리즘의 전체 시간 복잡도는  $\alpha((n_i + n_j)(1 + t_0 + n_i))$ 이다.

이 알고리즘의 장점은 분할 선을 매우 정확하게 추출해 내고, Voronoi 다이어그램을 이용하여 다각형의 골격[17]을 구하는 경우에 발생하는 곡선이 발생하지 않는다는데 있다. 또한 일반적으로 해법은 형태가 단순하므로 대부분의 경우 T-skeleton이 매우 효율적으로 얻어지며, 알고리즘의 구현도 비교적 간단한 장점이 있다. 또한 제안된 방법에서는 분할선의 레벨을 조절할 수 있는데, 이는 실제 물체에서는 모 등치선이나 가지등치선 레벨에서 분할이 시작되지 않고 두 층의 중간에서 분할이 시작될 가능성이 크므로 분할선 레벨을 두 층의 중간으로 사용함으로써 보다 실제 표면에 가까운 표면 모델을 만들 수 있게 된다. 또한 가장 흔히 발생하는 이중분기 문제를 다중분기 문제와 구분하여, 추가적인 가상의 등치선을 만들 필요가 없이 처리할 수 있다는 장점이 있다.

5. 다중 분기

앞장에서도 언급한 바와 같이 분기문제는 인접한 단층사이의 거리에 영향을 받게 되는데, 인접 단층간의 간격이 가까우면 대부분의 분기가 이중 분기의 형태로 나타나지만 물체의 크기에 비하여 단층사이의 간격이 큰 경우에는 다중 분기가 생길 가능성이 많아진다. 그림 8의 경우, 레벨  $z_k$ 의 등치선  $C_k^*$  이 레벨  $z_{k+1}$ 의 4개의 등치선과 연결되는 다중 분기  $\langle (C_k^*), (C_{k+1}^1, \dots, C_{k+1}^4) \rangle$ 가 발생하였는데, 만약  $z_k$ 와  $z_{k+1}$ 의 중앙레벨  $z_{k+0.5}$ 에서 한번 더 단층 샘플링이 이루어졌더라면 그림에서 보는 것과 같이 원래의 다중분기가 아닌  $z_k$ 와  $z_{k+0.5}$ 사이의 이중분기  $\langle (C_k^*), (C_{k+0.5}^1, C_{k+0.5}^2) \rangle$ 와,  $z_{k+0.5}$ 와  $z_{k+1}$ 사이의 두 개의 이중 분기  $\langle (C_{k+0.5}^1), (C_{k+1}^1, C_{k+1}^2) \rangle$ 와  $\langle (C_{k+0.5}^2), (C_{k+1}^3, C_{k+1}^4) \rangle$  문제가 된다. 본 논문의 다중 분기 알고리즘은 이러한 고찰에 근거한다. 다중분기가 발생한 경우(그림 8), 모 등치선과 가지 등치선 사이에 가상의 단층을 두어 다중 분기문제를 이중분기문제로 단순화시키는 것이다.

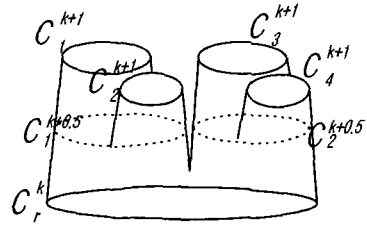


그림 8 다중분기와 가상층

모 등치선  $C_k^*$  이 인접한 층의  $N$ 개의 등치선과 표면으로 연결되어야 한다고 가정하고,  $BSet(C_k^*)$ 를  $C_k^*$ 에서 분기되는 가지등치선의 집합이라고 하자. 먼저  $BSet(C_k^*)$ 에서부터 병합할 한 쌍의 등치선  $(C_{k+1}^1, C_{k+1}^2)$ 을 골라내고 이를 병합할 가상의 등치선  $C_{k+0.5}^*$ 을 두 레벨 사이에서 만들어 이중분기  $\langle (C_{k+0.5}^*), (C_{k+1}^1, C_{k+1}^2) \rangle$ 를 해결하면, 모 등치선에서는  $C_{k+1}^1$  나  $C_{k+1}^2$ 와는 더 이상 직접 연결될 필요가 없으며, 대신 가상의 등치선  $C_{k+0.5}^*$ 와 표면으로 연결되어야 한다. 이것은  $BSet(C_k^*)$ 에서는  $C_{k+1}^1$ 와  $C_{k+1}^2$ 를 빼 내고 대신  $C_{k+0.5}^*$ 을 추가하는 것이 되며, 따라서 분기의 차수를 1 감소시키게 된다. 이 과정을  $BSet(C_k^*)$ 의 등치선 개수가 2가 될 때까지 계속 수행하면 모든 가지 등치선들은 가상의 레벨의 가상의 등치선에 이중분기로 연속적으로 연결되게 된다. 즉 다중 분기문제가 다수의 이중분기문제로 단순화되는 것이다. 이러한 접근방법은 인공물보다는 인체의 장기들과 같은 자연적인 물체에 대해 매우 자연스럽게 적용이 되며, 다중분기의 경우에도 매우 정교한 표면을 만들 수 있게 된다. 제안된 방법으로 다중분기를 처리하는 경우,  $BSet(C_k^*)$ 에서부터 병합 등치선 쌍을 선택하는 문제와 가상의 등치선을 만드는 방법이 중요한 사항이 된다.

5.1 병합 등치선 쌍(merging contour pair)의 선택

$N$ 중 분기문제  $\langle (C_k^*), (C_{k+1}^1, \dots, C_{k+1}^N) \rangle$ 가 주어졌을 때, 병합 등치선 쌍(merging contour pair)을 선택하는데는  $nC_2 \cdot n-1C_2 \cdot \dots \cdot 2C_2$  가지의 방법이 있는데, 등치선이 외의 다른 표면정보가 없으므로 적절한 휴리스틱을 사용하여 합병할 쌍을 선택하여야 한다. 예를 들어 가상 등치선의 면적을 최소화하는 방향으로 최적의 등치선 쌍을 선택하는 것도 하나의 해결방법이다. 그러나 이 경우 가지 등치선들 간의 형태적인 요인이 고려되지 않을 수도 있다. 예를 들어 그림 9의 (a)의 경우 가상 등치선의 면적을 최소화하기 위해서는  $(C_1, C_2)$ 나  $(C_2, C_3)$ 보다 면적이 작은  $(C_1, C_3)$ 를 병합해야 하겠지만 이 경우

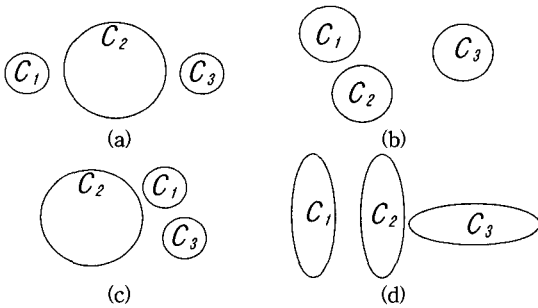


그림 9 다양한 다중분기의 가지 등치선들

매우 부자연스러운 표면이 만들어 지게 된다. 본 논문에서는 병합 등치선을 결정하기 위해 다음의 세 가지 우선 순위 조건을 정의하였다.

1. 지리적으로 가까운 등치선들을 먼저 합병하여야 한다. 가지 등치선들이 크기나 형태 또는 방향이 유사하다면, 가까운 등치선을 먼저 합하는 것이 당연할 것이다. 예를 들어 그림 9의 (b)의 경우  $(C_1, C_3)$ 나  $(C_2, C_3)$  보다는 면적이 작은  $(C_1, C_2)$ 를 병합하는 것이 좋다.

2. 작은 등치선을 가능한 한 먼저 합병하여야 한다. 그림 9의 (c)의 경우  $(C_1, C_2)$ 의 정점들이  $(C_2, C_3)$ 보다는 가까이 있지만, 면적이 상대적으로 작은  $(C_2, C_3)$ 를 먼저 병합하고 이것을  $C_1$ 과 합하는 것이 좋다. 이것은 가지 등치선의 크기를 가능한 한 고르게 만드는 효과가 있다.

3. 1과 2의 조건이 비슷한 경우, 두 등치선의 MER(Minimum Enclosing Rectangle)이 정사각형에 가까우면 먼저 합병하여야 한다. 이 조건은 만들어 지는 가상의 등치선의 형태를 길쭉하거나 하지 않고 균일하게 만드는 효과가 있다. 그림 9의 (d)의 경우  $(C_1, C_2)$ 를 먼저 병합하는 것이  $(C_2, C_3)$ 보다는 균일한 가상의 보간(interpolating) 등치선을 만들 수 있다.

이러한 고찰로부터 병합 등치선 쌍을 결정하기 위해 다음과 같이 인접 등치선 쌍(closest contour pair)을 정의하였다.

**Definition 1:**  $C_k^*$ 의 가지등치선의 집합을  $BSet(C_k^*) = \{C_1, C_2, \dots, C_N\}$  이라고 할 때,

$$d_{centroid}(C_i, C_j) = \text{MIN}(d_{centroid}(C_k, C_l)), 0 \leq k < N, 0 \leq l < N, k \neq l,$$

을 만족하는  $(C_i, C_j)$ 를 인접 등치선 쌍(closest contour pair)라 한다. 이때  $d_{centroid}(C_i, C_j)$ 는 두 등치선의 무게중심의 거리이며  $N$ 은 총 가지등치선의 개수이다.

본 논문에서는 이와 같이 정의된 인접 등치선 쌍을

찾아 병합하였는데, 이것은 앞에서 고찰한 세 가지 우선 순위 조건을 만족시키며, 실제로 다양한 분기에 대하여 실험한 결과 좋은 성능을 나타내었다.

일반적으로  $N$ 중 분기의 경우  $N-2$ 개의 가상 보간 등치선이 생기는데, 병합 등치선을 선택하는 순서에 따라 보간 등치선들이 같은 가상 레벨에 있을 수도 있고, 다른 레벨에 있을 수도 있다. 이러한 가상 레벨의 수는 최소  $\log_2 N$ 이고 최대  $N-2$ 가 된다. 그림 10은 4중 분기를 연속적인 이중분기로 병합하는 두 가지 방법을 보여주는데, 등치선의 증상을 연결한 그래프를 분기 그래프(branch graph)라 한다.(이것은 항상 트리형태가 아니고 때로는 사이클이 존재하므로 그래프이다) 왼쪽 그림과 같은 순서로 병합할 경우 2층의 가상 레벨이 생기며, 오른쪽과 같은 순서로 병합하는 경우에는 하나의 가상 레벨로 4중 분기를 해결할 수 있다. 이 방식으로 가지 등치선을 결정하는 알고리즘의 시간 복잡도는  $O(N^2)$ 인데, 등치선의 수  $N$ 이 일반적으로 작은 값이므로 실제 수행 시간은 무시할 만하다.

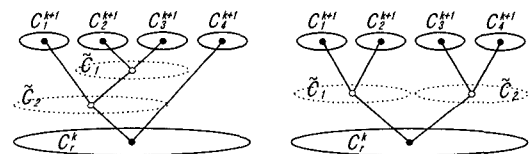


그림 10 4중 분기의 병합

가상의 보간 등치선(interpolating contour)의 레벨은 다음의 수식으로 계산이 된다.

$$z_c^i = z_c^k + (z_c^{k+1} - z_c^k) \frac{\text{depth}(C^k, C^i)}{\text{depth}(C^k, C_{\max}^{k+1})}$$

이때,  $\text{depth}(C^k, C^i)$ 는 두 등치선간의 거리(레벨 차)이며,  $\text{depth}(C^k, C_{\max}^{k+1})$ 는 모 등치선과 가지 등치선의 최대 거리를 나타낸다.

### 5.2 보간 등치선(interpolating contour)

등치선의 분기 그래프에서 중간 노드들이 보간 등치선들이 되며, 이것을 이용하여 두 개의 가지 등치선을 하나로 병합하게 된다. 본 논문에서는 이러한 보간 등치선으로 가지 등치선들의 블록껍질을 이용하였다. 즉 앞에서 해법을 구할 때와 같이 두 가지 등치선의 블록껍질을 구해 다리(bridge)를 찾고, 이 다리로 두 가지 등치선을 연결하여 하나의 보간 등치선을 만든다. 이러한 보간 등치선은 가지 등치선의 형태를 잘 반영하고 있으므로 매우 자연스러운 표면을 만들어 낼 수 있다.

경우에 따라서 보간 등치선의 크기가 모 등치선에 비해 너무 큰 경우에는 적당한 비율로 보간 등치선의 크기를 줄여 사용할 수도 있다. 그림 11은 보간등치선 생성 알고리즘이다.

```

Procedure Interpolating_Contour
Input: A merging contour pair  $C_u$  and  $C_v$ .
Output: An interpolating contour which branches  $C_u$  and  $C_v$ .
Begin
[1] Compute convex hull of the merging contours,  $C_u$  and  $C_v$ .
[2] If there are two common supporting segments(bridge), then construct the interpolating contour using the bridges and outer chains of  $C_u$  and  $C_v$ .
[3] Else find the handle and construct the interpolating contour using the handle and outer chain of  $C_u$  (or  $C_v$ ).
[4] Return the interpolating contour.
End
    
```

그림 11 보간 등치선 생성 알고리즘

**Theorem 2:** 알고리즘 Interpolating\_Contour의 시간 복잡도는  $O(n_u + n_v)$  이다. 이때  $n_u$  및  $n_v$  는 각각  $C_u$  와  $C_v$  의 정점의 개수이다.

**Proof:** Theorem 1에서 본 바와 같이 두 단순다각형의 합의 볼록껍질 계산의 시간 복잡도는  $O(n_u + n_v)$ 이므로 Step 1은 선형 시간만에 처리가 된다. Step 2와 Step 3도 볼록껍질의 정점수에 비례하는 시간안에 처리할 수 있으며, 볼록껍질의 정점의 개수는  $n_u + n_v$  이하이므로 이 과정의 시간복잡도 역시  $O(n_u + n_v)$ 이다. 따라서 알고리즘의 전체 시간복잡도는  $O(n_u + n_v)$ 이다.

**5.3 다중 분기 알고리즘**

그림 12는 다중분기 알고리즘을 보여주고 있다.

**Theorem 3:** 알고리즘 Multiple\_Branching은 시간 복잡도  $O(N^3 + N^2n)$ 에 N중 분기문제를 N-2개의 이중 분기 문제로 단순화시킨다. 이때  $N$  은 가지 등치선의 개수이며  $n$  은 등치선들의 정점의 총 개수이다.

**Proof:** Step 1은 상수시간에 처리된다. Step 2.1은  $O(N^2)$ 에 처리되며, Step 2.3은 Theorem 2에 의해  $O(n_u + n_v)$ 만에 처리된다. Step 2.3 및 2.4는 상수시간에 처리되고, While 루프가 N-2 번 수행되므로 Step 2 전체의 시간 복잡도는 최악의 경우  $O(N^3 + N^2n)$ 이다. Step 3 상수시간에, Step 4는  $O(Nn)$ 만에 처리가 된다. 따라서 알고리즘의 전체 시간 복잡도는  $O(N^3 + N^2n)$ 이다.

일반적으로 분기되는 등치선의 개수는 등치선의 정점

```

Procedure Multiple-Branching
Input: A root contour  $C_r^*$  at level  $z_r$  and  $n$  branch contours  $(C_1^{*+1}, \dots, C_n^{*+1})$  of  $C_r^*$  at level  $z_{r+1}$ .
Output: A list of double branching  $S_d$ .
Begin
[1] Initialization:  $BSet(C_r^*) \leftarrow \{C_1^{*+1}, \dots, C_n^{*+1}\}$ ,  $S_d \leftarrow \emptyset$ .
[2] While the number of the contours in  $BSet(C_r^*) > 2$ ,
    Do:
    [2.1] Find the merging contour pair  $(C_i, C_j)$  in  $BSet(C_r^*)$ .
    [2.2] Generate an interpolating contour  $C_{i+j}^{*+2}$  using the procedure Interpolating_Contour.
    [2.3] Insert the double branching  $\langle (C_i^{*+1}), (C_j, C_j) \rangle$  into  $S_d$ .
    [2.4] Replace  $(C_i, C_j)$  in  $BSet(C_r^*)$  with  $C_{i+j}^{*+2}$ .
    End while.
[3] Insert the double branching  $\langle (C_r^*), (C_1, C_2) \rangle$  into  $S_d$ .
[4] Set level of all the imaginary interpolating contours.
[5] Return  $S_d$ .
End
    
```

그림 12 다중 분기 알고리즘

에 수에 비하여 무시할 만 한 개수이므로, 이 알고리즘은 등치선의 정점의 수에 비례하는 시간에 주어진 다중 분기문제를 이중분기 문제로 단순화시킨다. 제안된 알고리즘은 인간의 심상이나 관철 등 인간의 장기 기관 등과 같이 등치선이 타원형에 가까울수록 더 잘 동작하며, 인간의 뇌와 같이 등치선의 형태가 복잡하게 꼬여있는 경우는 5.1절에서 언급한 바와 같은 방법으로 병합 등치선을 선택하는 것이 더 좋을 수도 있다.

**5.4 중복된 다중 분기**

앞에서 살펴본 이중분기 및 다중분기에서는 모 등치선이 다른 다중분기의 가지 등치선이 아닌 경우이다. 대부분의 분기는 이와 같이 일어나지만 경우에 따라서는 보다 더 복잡한 경우, 즉 모 등치선이 또 다른 다중분기의 가지 등치선인 경우도 발생한다. 이와 같은 경우는 인간의 뇌와 같이 복잡한 형태의 물체들에서 얻은 등치선에서 나타날 수 있다. 그림 13과 같은 분기 그래프의 경우, 등치선  $C_2, C_3$  및  $C_6$  는 각각 삼중분기의 모 등치선 이면서 또한 다중 분기의 가지 등치선이다. 이 경우 앞에서의 알고리즘을 바로 적용하는 경우 보간 등치선이 서로 겹치는 등 매우 부자연스러운 표면을 만들 수 있다. 제안된 방법으로 이러한 문제를 해결하기 위한 중요한 두 개의 단서는 (1) 제안된 이중분기 알고리즘은 가상의 등치선을 가상레벨에서 만들지 않기 때문에 가능한 한 먼저 적용되어야 한다는 것과, (2) 모 등치선이 다른 다중분기의 가지 등치선인 경우 가능한 한 나중에



병합되어야 한다는 것이다. 이러한 고찰에서부터 복잡한 분기 그래프는 다음과 같은 방법으로 해결할 수 있다.

P1: 모든 이중 분기를 먼저 단순화한다.

P2: 모 등치선이 가지 등치선이 아닌 모든 다중분기를 이중분기로 단순화한다.

P3: 분기가 남아있으면 다시 P1, P2를 반복하고, 없으면 종료한다.

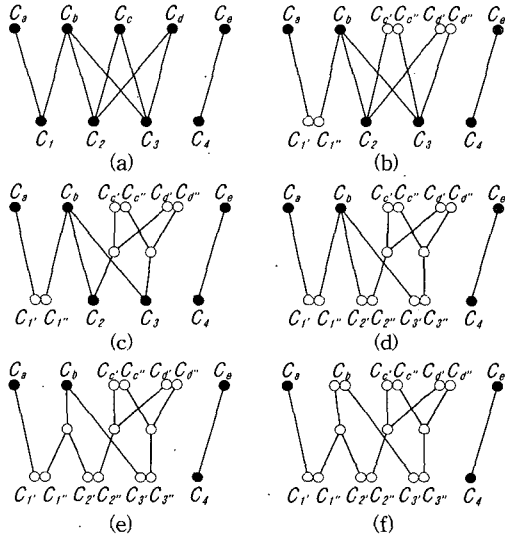


그림 13 복잡한 다중 분기 문제

그림 13은 제안된 방법으로 복잡한 분기문제를 해결하는 과정을 나타내고 있다. 먼저  $C_1$  및  $C_5$ ,  $C_d$ 로부터 분기되는 이중분기를 모두 해결한다(b). 이 과정에 의해 원래는 가지 등치선이었던 모 등치선  $C_2$  및  $C_3$ 가 더 이상 가지등치선이 아니므로, 이들에 제안된 다중분기 알고리즘을 적용하여 단순화하게 되는데, 결과적으로 이들은 이중분기의 모 등치선으로 변하게 된다(c). 이들에 다시 이중분기 알고리즘을 적용하면 분기 그래프는 (d)와 같이 단순화되고, 마지막으로 남은 다중분기인  $\langle(C_b), (C_{1'}, C_2, C_3)\rangle$ 에 다중분기 알고리즘을 적용하면 (e)가 되며 최종적으로  $C_b$ 를 모 등치선으로 하는 이중분기를 해결하면 (f)와 같이 완전히 단순화 된 분기 그래프가 된다.

제안된 방법은 기존의 분기 알고리즘에서 다루지 못했던 복잡한 분기문제를 처리할 수 있다는 장점이 있다. Meyers[12]는 이중분기 해결방법을 제안하였는데, 본 논문에서 정의한 해결과 유사한 계곡을 정의하고, 이를 분기 등치선 높이에서 바로 삼각화 해서 덮어버리는

Canyon tiling 알고리즘을 제안하였다. 그림 14의 (a)에서 회색 영역이 이들 알고리즘에 의해 덮인 계곡이 된다. 이 방법은 계곡의 끝자기를 충분히 표현하지 못하는데, 특히 그림과 같이 인접한 단층에서 연속적으로 분기가 생긴 경우 중간에 뚫린 영역을 표현할 수 없다. 이에 비하여 제안된 알고리즘은 계곡의 끝자기를 매우 정교하게 표현할 수 있고, 그림 14과 같이 연속적인 분기의 경우에도 (b)와 같이 훨씬 자연스러운 표면을 만들어 낸다는 장점이 있다. 또한 Meyers의 논문에는 다중분기에 대한 언급이 없는데 비하여, 제안된 알고리즘은 이중분기뿐 아니라 다중분기가 복잡하게 나타나 있는 경우 등, 일반적인 분기 문제에 대한 구조적인 해결 방법이 된다.

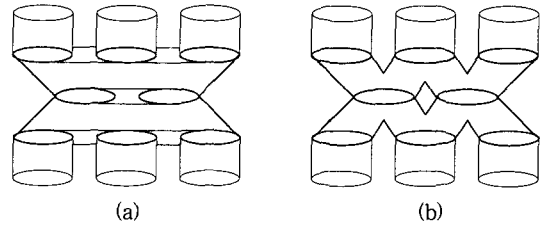


그림 14 Meyers의 방법(a)과 제안된 알고리즘(b)의 비교

### 6. 실험

제안된 알고리즘은 C++ 및 C언어를 이용하여 작성되었고, Sun Workstation 및 PC에서 구현되었다. 실험에는 인간의 단층 촬영 영상에서 직접 추출한 등치선 데이터와, 이중 및 다중 분기 알고리즘의 동작을 확인하기 위한 몇 개의 인공적인 등치선이 사용되었다.

먼저 그림 15의 (a)에서 2~3 레벨사이 및 4~5 레벨 사이에서 표면의 이중분기가 발생하였다. 이러한 이중분기 문제에 제안된 알고리즘을 적용하여 만들어진 분할선이 (b)에 나타나 있는데, 이것은 모 등치선을 매우 자연스럽게 분할하였으며, 결과적으로 이중분기를 두 개의 단일분기로 단순화하였다. 또한 주목할 것은 분할선의 레벨을 두 인접한 레벨의 중앙을 사용함으로써 (c) 및 (d)에서 볼 수 있듯이 매우 자연스러운 분기 표면을 만들어 내는 것으로, Meyers[12]나 Ekoule[11] 등 기존에 제안된 알고리즘으로는 이러한 표면을 만들기 어렵다. (c)는 모든 분기문제를 해결하여 단일분기화 된 등치선사이의 표면을 만들어 낸 결과이며, (d)는 평활명암법을 적용하여 가시화 한 결과이다.

그림 16에서는 원 등치선 (a) 및 (d)에 다중분기가

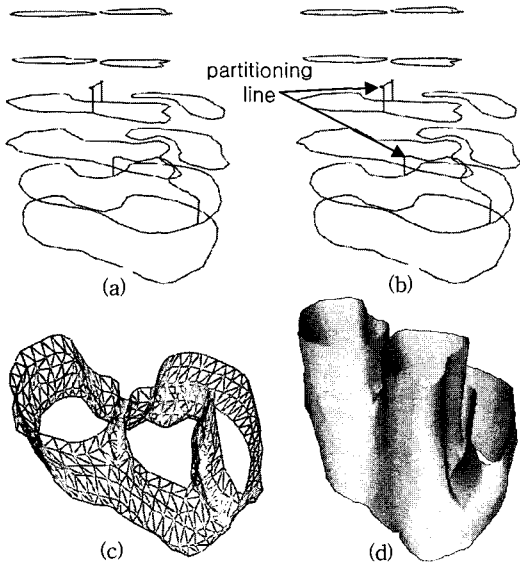


그림 15 이중 분기 등치선(double)

발생하였다. (a)의 경우 레벨 2와 3 사이에서 3중 분기가 발생하였고, (d)의 경우 5중 분기가 발생하였다. 이러한 분기 문제에 제안된 알고리즘을 적용하면 각각 (b)와 (f)와 같이 단순화되는데, (b)에서는 하나의 가상 보간 등치선이 만들어졌으며 이것으로 원래의 3중 분기문제를 2개의 이중분기로 단순화하였고, 각각에 이중분기 알고리즘을 적용하여 그림 (b)와 같은 분할선을 구하였다. (c)는 모든 분기문제가 해결된 (b)등치선에서 표면을 추출한 결과를 평활 명암법을 이용하여 가시화 한 영상이다. (d)등치선의 경우에는 그림 (e)와 같이 3개의 보간 등치선이 만들어져서 다섯 개의 원 가지 등치선들을 연속적으로 병합하였고 (f)는 결과적인 표면을 3차원으로 도시한 렌더링 영상이다. 이들에서 알 수 있듯이 제안된 이중분기 및 다중분기 알고리즘은 복잡한 분기 문제에 매우 효과적이며 자연스럽게 적용된다는 것을 알 수 있다.

그림 17의 (a)는 인간의 관절부분의 단층영상에서 추출한 등치선을 나타내며 (b), (c) 및 (d)는 각각 분기 해결후의 등치선, 표면 모델링 결과 및 평활 명암법에 의한 렌더링 결과영상을 보여주고 있다. 그림 18은 인간의 폐(lung) 부분을 단층 촬영하여 등치선을 추출한 결과를 보여주고 있는데, 제안된 알고리즘을 적용하여 표면모델을 구한 결과가 (b)에 나타나 있고, 표면모델을 두 가지 다른 방향에서 평활 명암법에 의해 렌더링한 결과를 (c)와 (d)에서 보여주고 있다.

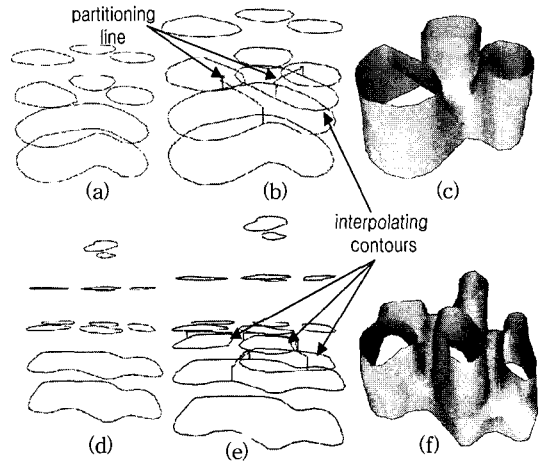


그림 16 다중 분기 등치선(multiple3, multiple5)

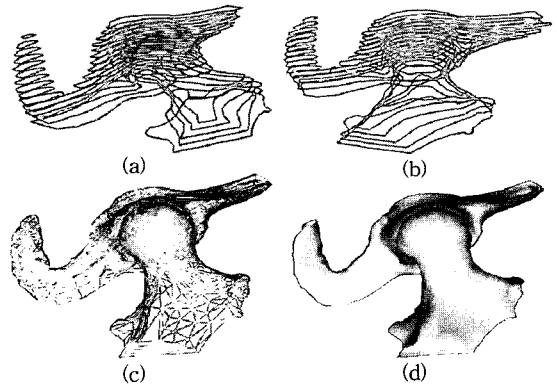


그림 17 hip 등치선

그림 19는 각 데이터에 대한 실험결과 데이터로 총 단층의 수, 원 등치성의 정점의 수, 모델링에 사용된 정점의 수, 표면 모델에서 삼각형의 수 및 각 분기가 나타난 횟수를 보여주고 있다. hip과 lung에서는 이중분기가 각각 1회, 5회 일어났으며, 다중분기 실험을 위한 인공적인 등치선에서 3중 분기를 2개의 이중분기로, 5중 분기를 4개의 이중분기로 간략화하여 해결하였다. 이중분기나 다중분기에서는 원 등치선의 정점 이외에 분할선이나 보간 등치선을 위해 추가로 정점들이 사용된다. 이들에 의해 표면조각의 수가 조금 증가하지만, 일반적인 데이터에서는 분기가 일어나는 층이 그렇게 많지 않고, 삼각 분할법 자체가 복셀 기법(voxel technique)에 비해 훨씬 적은 수의 표면조각을 만들어내므로 전체적인 표면조각의 수는 크게 증가되지는 않는다. 실험은

Pentium-II 400MHz급 PC에서 이루어졌으며, 분기를 처리하는데 걸리는 시간은 100msec이하였으며, 전체 모델링에 걸린 시간도 수 초 이하로 대부분의 시간이 단 일분기를 처리하는데 소요되었다. 이러한 실험들을 통해 제안된 알고리즘이 다양한 분기문제 해결에 매우 체계적인 해결책이며, 실제 데이터에 적용할 경우에도 자연스러운 표면을 만들어 내는 것을 알 수 있었다.

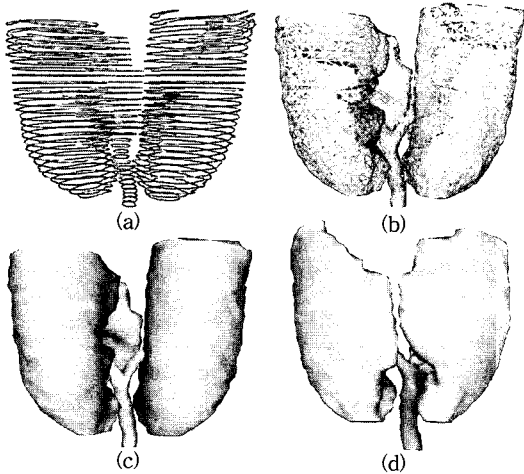


그림 18 lung 등치선

Data	Num. Slices	Num. Vertex (original)	Num. Vertex (model)	Num. Triangle	Num Double branch org(mod)	Num. Multiple Branch
Double	6	506	600	987	2	0
multiple3	4	399	498	784	0(2)	1(3중)
multiple5	5	310	617	1022	0(4)	1(5중)
hip	14	964	976	1747	1	0
lung	34	3120	3173	5927	5	0

그림 19 실험 데이터

### 7. 결론

본 논문에서는 등치선(wire-frame contour)정보로 표현된 물체의 볼륨정보에서부터 3차원 표면을 재구성하는 등치선 삼각화법(contour triangulation)에서 가장 문제가 되는 표면 분기문제(branching problem)를 체계적으로 해결하기 위한 방법을 제시하였다. 분기문제는 하나의 등치선이 인접한 층에 있는 두 개이상의 등치선과 연결되는 형태로 나타나는데, 표면 생성시 많은 모호성을 발생시킨다. 본 논문에서는 이러한 분기문제를 가

장 일반적으로 발생하는 형태인 이중분기와 그 이상의 다중분기 문제로 구분하여, 먼저 이중분기 알고리즘을 제안하였고, 다중분기문제를 다수의 이중분기문제로 단순화하는 다중분기 알고리즘을 제안하였다.

제안된 이중분기 알고리즘은 먼저 해협다각형을 구하고 이를 Toussiant의 방법[14]으로 삼각화하여 모 등치선의 분할선(partitioning line)을 구하는 것에 바탕을 두고 있다. 제안된 방법은 이중분기가 매우 복잡하게 나타나는 경우에도 잘 적용이 되며, 분할선의 레벨을 조절함으로써 매우 사실적인 표면을 만들어 낼 수 있었다. 본 논문에서는 다중분기문제를 단층 간격의 문제로 규정하고, 인접한 두 층 사이에 가상의 등치선을 만들고, 이를 이용하여 가지 등치선을 연속적으로 병합하는 방법을 사용하였는데, 모델링 대상물이 인공적으로 가공된 물체가 아니라 인체의 내부기관들과 같이 자연물일 경우 매우 적절한 접근방법이라 생각된다. 제안된 방법은 등치선 삼각화의 가장 큰 문제인 분기문제를 해결하기 위한 매우 구조적인 접근방법으로, 다양한 실제 등치선 데이터에 적용한 결과 좋은 성능을 나타냈다.

### 참고 문헌

- [1] K. Hahn and R. Bernstein, "Shading 3d-images from CT using gray-level gradients," *IEEE Trans. Medical Imaging*, pp. 45-47, 1986.
- [2] M. Levoy, "Display of surfaces from volume data," *IEEE Computer Graphics and Applications*, vol. 8, no. 3, pp. 29-37, 1988.
- [3] C. H. Chien and J. K. Aggarwal, "Volume/surface octrees for the representation of three-dimensional objects," *Computer Vision, Graphics, and Image Processing*, vol. 36, pp. 100-113, 1986.
- [4] G. Herman and H. Liu, "Three-dimensional display of human organs from computed tomograms," *Computer Graphics and Image Processing*, vol. 9, pp. 1-21, 1979.
- [5] W. E. Lorensen and H. E. Cline, "Marching cubes: a high resolution 3-d surface construction algorithm," *Comput. Graph.*, vol. 21, no. 4, pp. 163-169, 1987.
- [6] 최영규, 이의택, "셀 경계방식을 이용한 의료영상의 3차원 모델링" 정보과학회지 심사 중
- [7] H. Fuchs, F. Peyrin, and C. L. Odet, "A triangulation algorithm from arbitrary shaped multiple planar contours," *Communication of the ACM*, vol. 20, no. 10, pp. 693-702, Oct. 1977.
- [8] H. N. Christiansen and T. W. Sederberg, "Conversion of complex contour line definition into polygonal element mosaics," *Comput. Graph.*, vol.

- 20, pp. 693-702, 1978.
- [9] Ganapathy and T. G. Dennehy, "A new general triangulation method for planar contours," *Comput. Graph.*, vol. 16, no. 3, pp. 69-75, July 1982.
- [10] A. B. Ekoule, F. Peyrin, and C. L. Odet, "A triangulation algorithm from arbitrary shaped multiple planar contours," *ACM Transactions of Graphics*, vol. 10, no. 2, pp. 182-199, Apr. 1991.
- [11] Y. K. Choi, "Band Partitioning Algorithm for Surface Reconstruction from Planar Contours," *Electronics Letters*, accepted for publication.
- [12] D. Meyers, S. Skinner, and K. Sloan, "Surfaces from contours," *ACM Transactions of Graphics*, vol. 11, no. 3, pp. 228-258, July 1992.
- [13] J. D. Boissonnat, "Shape reconstruction from planar cross sections," *Computer Vision, Graphics, and Image Processing*, vol. 44, pp. 1-29, 1988.
- [14] G. Toussaint, "Efficient triangulation of simple polygons," *The Visual Computer*, vol. 7, no. 5-6, pp. 280-295, Sep. 1991.
- [15] 최영규, "의료영상의 3차원 Visualization을 위한 표면 모델링 기법의 연구 동향", 전기학회지, 제 45권, 8호, pp. 30-35, 1996년 8월.
- [16] D. T. Lee, "Medial axis transformation of a planar shape," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.4, no.4, pp. 363-369, July 1982.
- [17] F. P. Preparata and M. I. Shamos, *Computational Geometry, An Introduction*. Springer-Verlag, 1987.

#### 최 영 규



1985년 3월 ~ 1989년 2월 경북대학교 전자공학과 (학사). 1989년 3월 ~ 1991년 2월 한국과학기술원 전기 및 전자공학과 (석사). 1991년 3월 1995년 2월 한국과학기술원 전기 및 전자공학과 (박사). 1995년 3월 ~ 1999년 2월 LG산전

연구소 영상연구실 선임연구원. 1999년 3월 ~ 현재 한국기술교육대학교 정보기술공학부 전임강사. 관심분야는 Volume Visualization, 가상현실, Computer Vision, ITS 등